

Multi-view Discriminative Sequential Learning

Ulf Brefeld, Christoph Büscher, and Tobias Scheffer

Humboldt-Universität zu Berlin, Department of Computer Science,
Unter den Linden 6, 10099 Berlin, Germany
{brefeld, buescher, scheffer}@informatik.hu-berlin.de

Abstract. Discriminative learning techniques for sequential data have proven to be more effective than generative models for named entity recognition, information extraction, and other tasks of discrimination. However, semi-supervised learning mechanisms that utilize inexpensive unlabeled sequences in addition to few labeled sequences – such as the Baum-Welch algorithm – are available only for generative models. The multi-view approach is based on the principle of maximizing the consensus among multiple independent hypotheses; we develop this principle into a semi-supervised hidden Markov perceptron, and a semi-supervised hidden Markov support vector learning algorithm. Experiments reveal that the resulting procedures utilize unlabeled data effectively and discriminate more accurately than their purely supervised counterparts.

1 Introduction

The problem of labeling observation sequences has applications that range from language processing tasks such as named entity recognition, part-of-speech tagging, and information extraction to biological tasks in which the instances are often DNA strings. Traditionally, sequence models such as the hidden Markov model and variants thereof have been applied to the label sequence learning problem. Learning procedures for generative models adjust the parameters such that the joint likelihood of training observations and label sequences is maximized. By contrast, from the application point of view the true benefit of a label sequence predictor corresponds to its ability to find the correct label sequence given an observation sequence.

In the last years, conditional random fields [14, 15], hidden Markov support vector machines [4] and their variants have become popular; their discriminative learning procedures minimize criteria that are directly linked to their accuracy of retrieving the correct label sequence. In addition, kernel conditional random fields and hidden Markov support vector machines utilize kernel functions which enables them to learn in very high dimensional feature spaces. These features may also encode long-distance dependencies which cannot adequately be handled by first-order Markov models. Experiments uniformly show that discriminative models have advanced the accuracy that can be obtained for sequence labeling tasks; for instance, some of the top scoring systems in the BioCreative named entity recognition challenge used conditional random fields [18].

In the training process of generative sequence models, additional inexpensive and readily available unlabeled sequences can easily be utilized by employing Baum-Welch, a variant of the EM algorithm. But since EM uses generative models, it cannot directly be applied to discriminative learning. Text sequences are often described by high-dimensional attribute vectors that include, for instance, word features, letter n-grams, orthographical and many other features. These vectors can be split into two distinct, redundant views and thus the multi-view approach can be followed. Multi-view algorithms such as co-training [5] learn two initially independent hypotheses, and then minimize the disagreement of these hypotheses regarding the correct labels of the unlabeled data [11]. Thereby, they minimize an upper bound on the error rate [10].

The rest of our paper is structured as follows. Section 2 reports on related work and Section 3 reviews input output spaces and provides some background on multi-view learning. In Section 4 and 5 we present the dual multi-view hidden Markov kernel perceptron, and then leverage this algorithm to the multi-view hidden Markov support vector machine. We report on experimental results in Section 6. Section 7 concludes.

2 Related Work

In a rapidly developing line of research, many variants of discriminative sequence models are being explored. Recently studied variants include maximum entropy Markov models [17], conditional random fields [14], perceptron re-ranking [7], hidden Markov support vector machines [4], label sequence boosting [3], max-margin Markov models [21], case-factor diagrams [16], sequential Gaussian process models [2], kernel conditional random fields [15] and support vector machines for structured output spaces [22].

De Sa [11] observes a relationship between consensus of multiple hypotheses and their error rate and devises a semi-supervised learning method by cascading multi-view vector quantization and linear classification. A multi-view approach to word sense disambiguation combines a classifier that refers to the local context of a word with a second classifier that utilizes the document in which words co-occur [23]. Blum and Mitchell [5] introduce the co-training algorithm for semi-supervised learning that greedily augments the training set of two classifiers. A version of the AdaBoost algorithm boosts the agreement between two views on unlabeled data [9].

Dasgupta et al. [10] and Abney [1] give PAC bounds on the error of co-training in terms of the disagreement rate of hypotheses on unlabeled data in two independent views. This justifies the direct minimization of the disagreement. The co-EM algorithm for semi-supervised learning probabilistically labels all unlabeled examples and iteratively exchanges those labels between two views [20, 12]. Muslea et al. [19] extend co-EM for active learning and Brefeld and Scheffer [6] study a co-EM wrapper for the support vector machine.

3 Background

In this section we review “input output spaces” [2] and the consensus maximization principle that underlies multi-view algorithms for the reader’s convenience. In the remainder of our paper we adopt the clear notation proposed by [4].

3.1 Learning in Input Output Space

The setting of the *label sequence learning problem* is as follows. The labeled sample consists of n pairs $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)$, where $\mathbf{x}_i \in \mathcal{X}$ denotes the i -th input or observation sequence of length T_i ; *i.e.*, $\mathbf{x}_i = \langle x_{i,1}, x_{i,2}, \dots, x_{i,T_i} \rangle$, and $\mathbf{y}_i \in \mathcal{Y}$ the corresponding label sequence with $\mathbf{y}_i = \langle y_{i,1}, \dots, y_{i,T_i} \rangle$. We denote the set of all labels by Σ ; *i.e.*, $y_{i,t} \in \Sigma$.

In label sequence learning, joint features of the input and the label sequence play a crucial role (*e.g.*, “is the previous token labeled a named entity and both the previous and current token start with a capital letter?”). Such joint features of input and output cannot appropriately be modeled when the hypothesis is assumed to be a function from input to output sequences. The intuition of the input output space is that the decision function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ operates on a joint feature representation $\Phi(\mathbf{x}_i, \mathbf{y}_i)$ of input sequence \mathbf{x}_i and output sequence \mathbf{y}_i . Given an input, the classifier retrieves the output sequence

$$\hat{\mathbf{y}} = \operatorname{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}_i, \bar{\mathbf{y}}). \quad (1)$$

This step is referred to as decoding. Given the sample, the learning problem is to find a discriminator f that correctly decodes the examples. We utilize the \mathbf{w} -parameterized linear model $f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$. The joint feature representation $\Phi(\mathbf{x}, \mathbf{y})$ allows capturing non-trivial interactions of *label-label* pairs

$$\phi_{\sigma,\tau}(\mathbf{y}_i|t) = [[y_{i,t-s} = \sigma \wedge y_{i,t} = \tau]], \quad \sigma, \tau \in \Sigma, \quad (2)$$

($[[cond]]$ returns 1 if *cond* is true and 0 otherwise) and *label-observation* pairs

$$\bar{\phi}_{\sigma,j}(\mathbf{x}_i, \mathbf{y}_i|t) = [[y_{i,t} = \sigma]]\psi_j(x_{i,t-s}), \quad (3)$$

where many features $\psi_j(x_{i,t-s})$ extract characteristics of token $x_{i,t-s}$; *e.g.*, $\psi_{234}(x_{i,t-s})$ may be 1 if token $x_{i,t-s}$ starts with a capital letter and 0 otherwise. We will refer to the vector $\psi(x) = (\dots, \psi_j(x), \dots)^\top$ and denote the dot product by means of $k(x, \bar{x}) = \langle \psi(x), \psi(\bar{x}) \rangle$.

The feature representation $\Phi(\mathbf{x}_i, \mathbf{y}_i)$ of the i -th sequence is defined as the sum of all feature vectors $\bar{\Phi}(\mathbf{x}_i, \mathbf{y}_i|t) = (\dots, \phi_{\sigma,\tau}(\mathbf{y}_i|t), \dots, \bar{\phi}_{\sigma,j}(\mathbf{x}_i, \mathbf{y}_i|t), \dots)^\top$ extracted at time t

$$\bar{\Phi}(\mathbf{x}_i, \mathbf{y}_i) = \sum_{t=1}^{T_i} \bar{\Phi}(\mathbf{x}_i, \mathbf{y}_i|t). \quad (4)$$

Restricting the possible features to consecutive label-label (Equation 2 with $s = 1$) and label-observation (Equation 3 with $s = 0$) dependencies is essentially

a first-order Markov assumption and as a result, decoding (Equation 1) can be performed by a Viterbi algorithm in time $\mathcal{O}(T|\Sigma|^2)$, with transition matrix $A = \{a_{\sigma,\tau}\}$ and observation matrix $B_{\mathbf{x}} = \{b_{s,\sigma}(\mathbf{x})\}$ given by

$$a_{\sigma,\tau} = \sum_{i,\bar{\mathbf{y}}} \alpha_i(\bar{\mathbf{y}}) \sum_t [[\bar{y}_{t-1} = \sigma \wedge \bar{y}_t = \tau]] \quad (5)$$

$$b_{s,\sigma}(\mathbf{x}) = \sum_{i,t,\bar{\mathbf{y}}} [[\bar{y}_t = \sigma]] \alpha_i(\bar{\mathbf{y}}) k(x_s, x_{i,t}). \quad (6)$$

We utilize a kernel function $K((\mathbf{x}, \mathbf{y}), (\bar{\mathbf{x}}, \bar{\mathbf{y}})) = \langle \Phi(\mathbf{x}, \mathbf{y}), \Phi(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \rangle$ to compute the inner product of two observation and label sequences in input output space. The inner product decomposes into

$$\langle \Phi(\mathbf{x}, \mathbf{y}), \Phi(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \rangle = \sum_{s,t} [[y_{s-1} = \bar{y}_{t-1} \wedge y_s = \bar{y}_t]] + \sum_{s,t} [[y_s = \bar{y}_t]] k(x_s, \bar{x}_t). \quad (7)$$

3.2 The Consensus Maximization Principle

In the multi-view setting that we discuss here the available attributes \mathcal{X} are decomposed into disjoint sets \mathcal{X}^1 and \mathcal{X}^2 . An example $(\mathbf{x}_i, \mathbf{y}_i)$ is therefore viewed as $(\mathbf{x}_i^1, \mathbf{x}_i^2, \mathbf{y}_i)$, where $\mathbf{x}_i^v \in \mathcal{X}^v$, with $v = 1, 2$.

A characteristic of multi-view methods is the natural inclusion of unlabeled examples $(\mathbf{x}_1^1, \mathbf{x}_1^2), \dots, (\mathbf{x}_m^1, \mathbf{x}_m^2)$ which leads directly to semi-supervised techniques. Dasgupta et al. [10] have studied the relation between the consensus of two independent hypotheses and their error rate. One of their results that holds under some mild assumptions is the inequality

$$P(f^1 \neq f^2) \geq \max\{P(\text{err}(f^1)), P(\text{err}(f^2))\}. \quad (8)$$

That is, the probability of a disagreement of two independent hypotheses upper bounds the error rate of either hypothesis. Thus, the strategy of semi-supervised multi-view learning is: Minimize the error for labeled examples and maximize the agreement for unlabeled examples.

In the following the set D^l contains n labeled examples $(\mathbf{x}_i^1, \mathbf{x}_i^2, \mathbf{y}_i)$, $i = 1, \dots, n$, and D^u consists of m unlabeled sequences $(\mathbf{x}_i^1, \mathbf{x}_i^2)$, $i = n+1, \dots, n+m$, where in general $n < m$ holds.

4 Multi-view Hidden Markov Perceptrons

In this section we present the dual multi-view hidden Markov perceptron algorithm. For the reader's convenience, we briefly review the single-view hidden Markov perceptron [8, 4] and extend it to semi-supervised learning.

The Hidden Markov Perceptron

The goal is to learn a linear discriminant function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ given by

$$f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle, \quad (9)$$

that correctly decodes any example sequence $(\mathbf{x}_i, \mathbf{y}_i) \in D$; *i.e.*,

$$\mathbf{y}_i = \underset{\bar{\mathbf{y}}}{\operatorname{argmax}} f(\mathbf{x}_i, \bar{\mathbf{y}}). \quad (10)$$

Equation 9 can be transformed into its equivalent dual formulation given by

$$f(\mathbf{x}, \mathbf{y}) = \sum_i \sum_{\bar{\mathbf{y}}} \alpha_i(\bar{\mathbf{y}}) \langle \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}, \mathbf{y}) \rangle, \quad (11)$$

where the relation $\mathbf{w} = \sum_i \sum_{\bar{\mathbf{y}}} \alpha_i(\bar{\mathbf{y}}) \Phi(\mathbf{x}_i, \bar{\mathbf{y}})$ is used. The dual depends only on the inner product in input output space that can be computed efficiently by means of a kernel (Equation 7) and dual variables $\alpha_i(\bar{\mathbf{y}}) \in \mathbb{Z}$. The latter weight the importance of sequence $\bar{\mathbf{y}}$ for the prediction of observation \mathbf{x}_i .

The dual perceptron algorithm consecutively decodes each input in the training sample. When the decoding (Equation 11) yields an incorrectly labeled sequence $\hat{\mathbf{y}}$ for the i -th example, instead of the correct sequence \mathbf{y}_i , then the corresponding α_i are updated according to

$$\alpha_i(\mathbf{y}_i) = \alpha_i(\mathbf{y}_i) + 1; \quad \alpha_i(\hat{\mathbf{y}}) = \alpha_i(\hat{\mathbf{y}}) - 1. \quad (12)$$

Thus, after an error has occurred, the correct sequence receives more, the incorrect prediction receives less influence. Since all initial $\alpha_i = 0$ it suffices to store only those sequences in memory that have been used for an update.

The Multi-view Hidden Markov Perceptron

We now have labeled examples $(\mathbf{x}_i^1, \mathbf{x}_i^2, \mathbf{y}_i) \in D^l$ and unlabeled examples $(\mathbf{x}_i^1, \mathbf{x}_i^2) \in D^u$, where $\psi^1(x_{i,t}^1)$ and $\psi^2(x_{i,t}^2)$, $t = 1, \dots, T_i$, live in distinct vector spaces. We have decision functions $f(\mathbf{x}^1, \mathbf{x}^2, \mathbf{y}) = f^1(\mathbf{x}^1, \mathbf{y}) + f^2(\mathbf{x}^2, \mathbf{y})$ with

$$f^v(\mathbf{x}^v, \mathbf{y}) = \sum_{i=1}^{n+m} \sum_{\bar{\mathbf{y}}} \alpha_i^v(\bar{\mathbf{y}}) \langle \Phi^v(\mathbf{x}_i^v, \bar{\mathbf{y}}), \Phi^v(\mathbf{x}^v, \mathbf{y}) \rangle, \quad v = 1, 2. \quad (13)$$

According to the consensus maximization principle, the perceptron algorithm now has to minimize the number of errors for labeled examples and the disagreement for unlabeled examples. Each view $v = 1, 2$ predicts the label sequence for an example i , whether it is labeled or unlabeled, analogously to the single-view hidden Markov perceptron according to

$$\hat{\mathbf{y}}^v = \underset{\bar{\mathbf{y}}}{\operatorname{argmax}} f^v(\mathbf{x}_i^v, \bar{\mathbf{y}}). \quad (14)$$

The hidden Markov perceptron update rule for labeled examples remains unchanged; if view v misclassifies the i -th labeled example ($\mathbf{y}_i \neq \hat{\mathbf{y}}^v$), then the respective parameters are updated according to Equation 15.

$$\alpha_i^v(\mathbf{y}_i) = \alpha_i^v(\mathbf{y}_i) + 1; \quad \alpha_i^v(\hat{\mathbf{y}}^v) = \alpha_i^v(\hat{\mathbf{y}}^v) - 1. \quad (15)$$

Table 1. Multi-view HM perceptron algorithm

Input: n labeled sequences D^l , m unlabeled sequences D^u , number of iterations t_{max} .

-
1. Initialize all $\alpha_i^v(\mathbf{y}_i) = 0$, $v = 1, 2$.
 2. **For** $t = 1, \dots, t_{max}$: **For** all sequences $i = 1, \dots, n + m$
 3. Viterbi decoding: retrieve $\hat{\mathbf{y}}_i^1$ and $\hat{\mathbf{y}}_i^2$ (Equation 14).
 4. **If** i -th sequence is a labeled example and $\mathbf{y}_i \neq \hat{\mathbf{y}}_i^v$
 then update $\alpha_i^v(\cdot)$ according to Equation 15, $v = 1, 2$.
 5. **Elseif** i -th sequence is an unlabeled example and $\hat{\mathbf{y}}_i^1 \neq \hat{\mathbf{y}}_i^2$
 then update both views according to Equation 16.
 6. **End if.**
 7. **End for** i ; **End For** t .
-

Output: Combined hypothesis $f(\mathbf{x}^1, \mathbf{x}^2, \mathbf{y})$.

If the views disagree on an unlabeled example – that is, $\hat{\mathbf{y}}^1 \neq \hat{\mathbf{y}}^2$ – updates have to be performed that reduce the discord. Intuitively, each decision is swayed towards that of the peer view in Equation 16.

$$\alpha_j^v(\hat{\mathbf{y}}^{\bar{v}}) = \alpha_j^v(\hat{\mathbf{y}}^{\bar{v}}) + C_u; \quad \alpha_j^v(\hat{\mathbf{y}}^v) = \alpha_j^v(\hat{\mathbf{y}}^v) - C_u, \quad v = 1, 2. \quad (16)$$

The parameter $0 \leq C_u \leq 1$ determines the influence of a single unlabeled example. If $C_u = 1$ each example has the same influence whether it is labeled or unlabeled. The output $\hat{\mathbf{y}}$ of the joint decision function

$$\hat{\mathbf{y}} = \underset{\bar{\mathbf{y}}}{\operatorname{argmax}} f(\mathbf{x}^1, \mathbf{x}^2, \bar{\mathbf{y}}) = \underset{\bar{\mathbf{y}}}{\operatorname{argmax}} [f^1(\mathbf{x}^1, \bar{\mathbf{y}}) + f^2(\mathbf{x}^2, \bar{\mathbf{y}})] \quad (17)$$

can be efficiently computed by a Viterbi decoding. Viterbi needs a transition cost matrix that details the score of a label transition and an observation cost matrix that relates labels to observations. These quantities can be derived by summing the scores of the corresponding single-view matrices. The transition and observation matrices are given by $A = A^1 + A^2$ and $B = B^1 + B^2$, where $A^v = \{a_{\sigma, \tau}^v\}$ is defined in Equation 5 and $B_{\mathbf{x}}^v = \{b_{s, \sigma}^v(\mathbf{x}^v)\}$ in Equation 6, $v = 1, 2$, respectively. Table 1 shows the multi-view hidden Markov perceptron algorithm.

5 Multi-view Hidden Markov Support Vector Machines

In this Section we present the 1-norm and 2-norm multi-view hidden Markov SVMs. We omit the superscript for view $v = 1, 2$ and use the superscript \bar{v} to indicate variables of the peer view.

The aim in discriminative sequential learning is to learn f such that correct label sequences obtain higher scores than any other label sequence (Equation 18). The corresponding extension to unlabeled sequences is given in Equation 19 where the prediction of the peer view is treated as true label sequence.

$$f(\mathbf{x}_i, \mathbf{y}_i) - \max_{\bar{\mathbf{y}} \neq \mathbf{y}_i} f(\mathbf{x}_i, \bar{\mathbf{y}}) > 0, \quad i = 1, \dots, n \quad (18)$$

$$f(\mathbf{x}_i, \mathbf{y}^{\bar{v}}) - \max_{\bar{\mathbf{y}} \neq \mathbf{y}^{\bar{v}}} f(\mathbf{x}_i, \bar{\mathbf{y}}) > 0, \quad i = n+1, \dots, n+m. \quad (19)$$

The margin of the i -th sequence is defined as $\gamma_i^v = \max\{0, f^v(\mathbf{x}_i^v, \mathbf{y}_i) - \max_{\bar{\mathbf{y}} \neq \mathbf{y}_i} f^v(\mathbf{x}_i^v, \bar{\mathbf{y}})\}$ in views $v = 1, 2$. Support vector machines enforce confident predictions by maximizing the margin $\frac{1}{\|\mathbf{w}\|}$; this leads us to a hard margin optimization problem for each view.

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \forall_{i=1}^n, \quad \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1 \\ & \forall_{i=n+1}^{n+m}, \quad \forall_{\bar{\mathbf{y}} \neq \mathbf{y}^{\bar{v}}} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_j, \mathbf{y}^{\bar{v}}) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1. \end{aligned} \quad (20)$$

The constraints can be integrated into the objective by means of Lagrange multipliers $\alpha_i(\bar{\mathbf{y}})$ for each example i and each pseudo sequence $\bar{\mathbf{y}} \neq \mathbf{y}_i$ (here, the $\alpha_i(\bar{\mathbf{y}})$ weight the influence of the difference vector $\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}})$),

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n+m} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) - \frac{1}{2} \sum_{i,j=1}^{n+m} \sum_{\substack{\bar{\mathbf{y}} \neq \mathbf{y}_i \\ \bar{\mathbf{y}}' \neq \mathbf{y}_j}} \alpha_i(\bar{\mathbf{y}}) \alpha_j(\bar{\mathbf{y}}') K'_{i\bar{\mathbf{y}}, j\bar{\mathbf{y}}'} \\ \text{s.t.} \quad & \forall_{i=1}^{n+m} \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \quad \alpha_i(\bar{\mathbf{y}}) \geq 0, \end{aligned} \quad (21)$$

where we use $K'_{i\bar{\mathbf{y}}, j\bar{\mathbf{y}}'}$ shorthand for

$$K'_{i\bar{\mathbf{y}}, j\bar{\mathbf{y}}'} = \langle \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}_j, \mathbf{y}_j) - \Phi(\mathbf{x}_j, \bar{\mathbf{y}}') \rangle \quad (22)$$

$$= K((\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_j, \mathbf{y}_j)) - K((\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_j, \bar{\mathbf{y}}')) \quad (23)$$

$$- K((\mathbf{x}_i, \bar{\mathbf{y}}), (\mathbf{x}_j, \mathbf{y}_j)) + K((\mathbf{x}_i, \bar{\mathbf{y}}), (\mathbf{x}_j, \bar{\mathbf{y}}')). \quad (24)$$

Table 2. Working set optimization for labeled examples [4]

Input: i -th labeled sequence $(\mathbf{x}_i^1, \mathbf{x}_i^2, \mathbf{y}_i)$, $C > 0$, view $v \in \{1, 2\}$.

-
1. **Loop**
 2. compute $\hat{\mathbf{y}}^v = \operatorname{argmax}_{\mathbf{y} \neq \mathbf{y}_i} f^v(\mathbf{x}_i^v, \mathbf{y})$
 3. **If** $f^v(\mathbf{x}_i^v, \mathbf{y}_i) - f^v(\mathbf{x}_i^v, \hat{\mathbf{y}}^v) \geq 1$ **then** return α_i^v .
 4. **Else** $S^v = S^v \cup \{\hat{\mathbf{y}}^v\}$.
 5. Optimize $\alpha_i^v(\bar{\mathbf{y}})$ over $\Phi(\mathbf{x}_i^v, \mathbf{y}_i) - \Phi(\mathbf{x}_i^v, \bar{\mathbf{y}})$, $\forall \bar{\mathbf{y}} \in S^v$
 6. $\forall \mathbf{y} \in S^v$ with $\alpha_i^v(\mathbf{y}) = 0$: $S^v = S^v \setminus \{\mathbf{y}\}$
 7. **End if.**
 8. **End loop.**
-

Output: Optimized α_i^v .

In general, we have to allow pointwise relaxations of the hard margin constraint by slack variables leading us to a soft-margin optimization problem for each view,

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{r} \left(\sum_{i=1}^n \xi_i^r + C_u \sum_{i=n+1}^{n+m} (\min\{\gamma_i^{\bar{v}}, 1\}) \xi_i^r \right) \\ \text{s.t.} \quad & \forall_{i=1}^n, \quad \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1 - \xi_i \\ & \forall_{i=n+1}^{n+m}, \quad \forall_{\bar{\mathbf{y}} \neq \mathbf{y}^{\bar{v}}} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}^{\bar{v}}) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1 - \xi_i \\ & \forall_{i=1}^{n+m} \quad \xi_i \geq 0, \end{aligned} \quad (25)$$

where $r = 1, 2$ denotes a linear or quadratic penalization of the error, respectively, $C > 0$ determines the trade-off between margin maximization and error minimization, and C_u is a balancing factor that regularizes the influence of the unlabeled data. Weights of $\min\{\gamma_i^{\bar{v}}, 1\}$ to the slacks $\xi_{n+1}, \dots, \xi_{n+m}$ relate errors on unlabeled examples to the confidence of the peer view's prediction.

In case of a linear loss – *i.e.*, $r = 1$ – the inclusion of slack variables, costs, and balancing factor resolves into $n + m$ additional constraints of optimization problem 21 that upper bound the sum of the α_i .

$$\forall_{i=1}^n : \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \leq C; \quad \forall_{i=n+1}^{n+m} : \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i^{\bar{v}}} \alpha_i(\bar{\mathbf{y}}) \leq (\min\{\gamma_i^{\bar{v}}, 1\}) C_u C. \quad (26)$$

The necessary changes to optimization problem 21 in case of a quadratic penalty ($r = 2$) can be incorporated into the kernel by $K''_{i\bar{\mathbf{y}}, j\bar{\mathbf{y}}'} = K^1_{i\bar{\mathbf{y}}, j\bar{\mathbf{y}}'} + \Delta_{i\bar{\mathbf{y}}, j\bar{\mathbf{y}}'}$ where

$$\Delta_{i\bar{\mathbf{y}}, j\bar{\mathbf{y}}'} = \begin{cases} \frac{1}{C} & i = j, \bar{\mathbf{y}} = \bar{\mathbf{y}}', 1 \leq i, j \leq n \\ \frac{1}{(\min\{\gamma_j^{\bar{v}}, 1\}) C_u C} & i = j, \bar{\mathbf{y}} = \bar{\mathbf{y}}', n+1 \leq i, j \leq n+m \\ 0 & \text{otherwise.} \end{cases} \quad (27)$$

Table 3. Working set optimization for unlabeled examples

Input: i -th unlabeled sequence $(\mathbf{x}_i^1, \mathbf{x}_i^2)$, $C, C_u > 0$, repetitions r_{max} .

-
1. $S^1 = S^2 = \emptyset, \alpha_i^1 = \alpha_i^2 = 0$.
 2. **Loop**
 3. compute $\hat{\mathbf{y}}^1 = \operatorname{argmax}_{\mathbf{y}} f^1(\mathbf{x}_i^1, \mathbf{y})$ and $\hat{\mathbf{y}}^2 = \operatorname{argmax}_{\mathbf{y}} f^2(\mathbf{x}_i^2, \mathbf{y})$
 4. **If** $\hat{\mathbf{y}}^1 = \hat{\mathbf{y}}^2$ **then** return α_i^1 and α_i^2 .
 5. **Else For** $v = 1, 2$:
 6. Substitute former target: $\hat{\mathbf{y}}_i = \hat{\mathbf{y}}^{\bar{v}}$.
 7. Add pseudo sequence: $S^v = S^v \cup \{\hat{\mathbf{y}}^v\}$
 8. Optimize $\alpha_i^v(\bar{\mathbf{y}})$ over $\Phi(\mathbf{x}_i^v, \hat{\mathbf{y}}_i) - \Phi(\mathbf{x}_i^v, \bar{\mathbf{y}}), \forall \bar{\mathbf{y}} \in S^v$
 9. $\forall \bar{\mathbf{y}} \in S^v$ with $\alpha_i^v(\bar{\mathbf{y}}) = 0$: $S^v = S^v \setminus \{\bar{\mathbf{y}}\}$
 10. **End for** v . **End if.**
 11. **Until** consensus or r_{max} repetitions without consensus.
-

Output: Optimized α_i^1 and α_i^2 .

Since the dual variables $\alpha_i(\bar{\mathbf{y}})$ are tied to observation sequences \mathbf{x}_i , the optimization problem (Equation 21) splits into $n + m$ disjoint subspaces spanned by $\alpha_i(\cdot)$ with fixed values for the $\alpha_{j \neq i}(\cdot)$; the optimization iterates over these subspaces.

In an outer loop, the Hidden Markov SVM iterates over the examples and consecutively optimizes the example’s parameters $\alpha_i(\cdot)$, using distinct working set approaches for labeled (Table 2) and unlabeled (Table 3) data. Difference vectors $\bar{\mathbf{y}}$ with $\alpha_i(\bar{\mathbf{y}}) = 0$ are removed in order to speed up computation. When the loop reaches an unlabeled sequence, all pseudo sequences $\alpha_i(\cdot)$ of that example are removed since the disagreements that they used to correct in earlier iterations of the main loop may have been resolved.

Since the cost factors upper-bound the growth of the α_i for the 1-norm machine, consensus might not be established and we therefore integrate a user defined constant r_{max} that bounds the number of iterations. Linear Viterbi decoding can be performed similarly to Equation 5 and Equation 6.

6 Empirical Results

We concentrate on named entity recognition (NER) problems. We use the data set provided for task 1A of the BioCreative challenge and the Spanish news wire article corpus of the shared task of CoNLL 2002.

The BioCreative data contains 7500 sentences from biomedical papers; gene and protein names are to be recognized. View 1 consists of the token itself together with letter 2, 3 and 4-grams; view 2 contains surface clues like capitalization, inclusion of Greek symbols, numbers, and others as documented in [13]. The CoNLL2002 data contains 9 label types which distinguish person, organization, location, and other names. We use 3100 sentences of between 10 and 40 tokens which we represent by a token view and a view of surface clues.

In each experiment we draw a specified number of (labeled and unlabeled) training and holdout sentences without replacement at random in each iteration. We assure that each label occurs at least once in the labeled training data; otherwise, we discard and draw again. Each holdout set consists of 500 (BioCreative) and 300 (Spanish news wire) sentences, respectively. We first optimize parameter C_u using resampling; we then fix C_u and present curves that show the average token-based error over 100 randomly drawn training and holdout sets. The baseline methods (single-view HM perceptron and HM SVM) are trained on concatenated views; errorbars indicate standard error.

We use Alex Smola’s Loqo implementation as QP solver and initialize $r_{max} = 10$, $C = 1$. We employ a constant C_u for multi-view perceptron and use an exponential scheme to increase C_u to its maximal value in the 30th iteration. We want to answer the following questions.

Is the inclusion of unlabeled data beneficial for sequential learning?

Figure 1 shows learning curves for single-view and multi-view HM perceptron and HM SVM for both problems. With the exception of one point, the multi-view methods always outperform their single-view, purely supervised

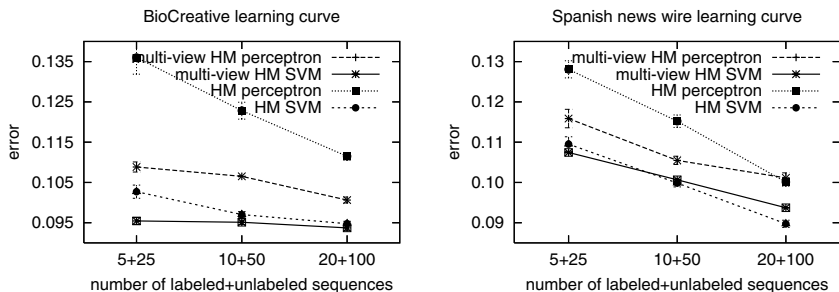


Fig. 1. Learning curves for BioCreative and Spanish news wire

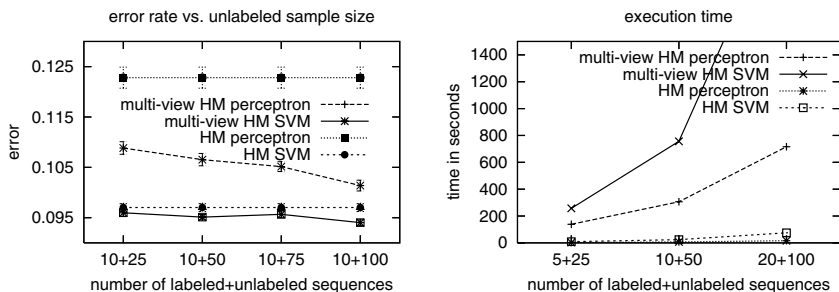


Fig. 2. Left: Error depending on the unlabeled sample size for BioCreative. Right: Execution time.

counterparts significantly; the multi-view HM SVM is the most accurate sequence learning method. We use a regular HMM as an additional baseline; its error rates of 23.59%, 20.04%, and 15.31% for 5, 10, and 20 training sequences for the news wire and 17.98%, 14.31%, and 12.31% (5, 10, 20 training sequences) for the BioCreative data lie above the plotted range of Figure 1. In Figure 2 (left) we vary the number of unlabeled sequences for the BioCreative data set. As the number of unlabeled data increases, the advantage of multi-view over single-view sequence learning increases further.

How costly is the training process?

Figure 2 (right) plots execution time against training set size. The performance benefits are at the cost of significantly longer training processes. The multi-view HM perceptron scales linearly and the multi-view HM SVM quadratically in the number of unlabeled sequences.

Are there better ways of splitting the features into views?

We compare the feature split into the token itself and letter n -grams versus surface clues to the average of 100 random splits. Surprisingly, Figure 3 shows that random splits work even (significantly) better. We also construct a feature split in which view 1 contains all odd, and view 2 all even features. Hence, each view contains half of the Boolean token features as well as half of the surface

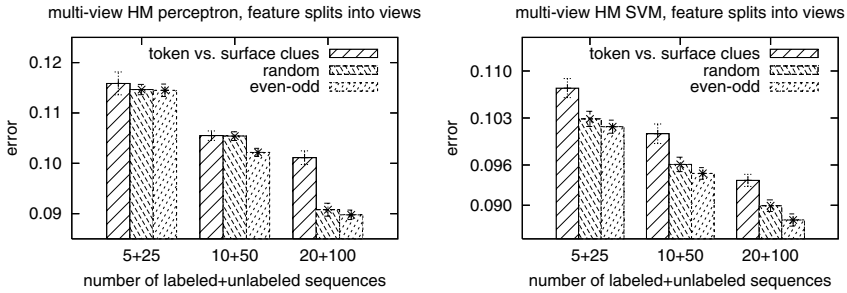


Fig. 3. Error for several splits of features into views for Spanish news wire

clues. Figure 3 shows that this split performs slightly but significantly better than the random split. Hence, our experiments show that even though multi-view learning using the split of token and n -grams versus surface clues leads to a substantial improvement over single-view learning, a random or odd-even split lead to an even better performance.

7 Conclusion

Starting from two discriminative sequence learning algorithms – the Hidden Markov perceptron and SVM – we constructed semi-supervised learning methods by utilizing the principle of consensus maximization between hypotheses. We derived the multi-view HM perceptron as well as multi-view 1-norm and 2-norm HM SVMs. Our experiments show that, on average, these methods utilize unlabeled data effectively and outperform their purely supervised counterparts significantly; the multi-view HM SVM achieves the highest performance.

We observed that random feature splits perform better than splitting the features into a token view and a view of surface clues. Nevertheless, the multi-view algorithms outperform their supervised counterparts even for the initial weak split. Our future work will address the construction of good feature splits.

Acknowledgment

This work has been funded by the German Science Foundation DFG under grant SCHE540/10-1.

References

1. S. Abney. Bootstrapping. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002.
2. Y. Altun, T. Hofmann, and A. J. Smola. Gaussian process classification for segmenting and annotating sequences. In *Proceedings of the International Conference on Machine Learning*, 2004.

3. Y. Altun, M. Johnson, and T. Hofmann. Discriminative learning for label sequences via boosting. In *Advances in Neural Information Processing Systems*, 2003.
4. Y. Altun, I. Tschantaris, and T. Hofmann. Hidden Markov support vector machines. In *Proc. of the International Conference on Machine Learning*, 2003.
5. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of the Conference on Computational Learning Theory*, 1998.
6. U. Brefeld and T. Scheffer. Co-em support vector learning. In *Proceedings of the International Conference on Machine Learning*, 2004.
7. M. Collins. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002.
8. M. Collins and N. Duffy. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems*, 2002.
9. M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.
10. S. Dasgupta, M. Littman, and D. McAllester. PAC generalization bounds for co-training. In *Proceedings of Neural Information Processing Systems*, 2001.
11. V. de Sa. Learning classification with unlabeled data. In *Proceedings of Neural Information Processing Systems*, 1994.
12. R. Ghani. Combining labeled and unlabeled data for multiclass text categorization. In *Proceedings of the International Conference on Machine Learning*, 2002.
13. J. Hakenberg, S. Bickel, C. Plake, U. Brefeld, H. Zahn, L. Faulstich, U. Leser, and T. Scheffer. Systematic feature evaluation for gene name recognition. *BMC Bioinformatics*, 6(1):S9, 2005.
14. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.
15. J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *Proc. of the Int. Conference on Machine Learning*, 2004.
16. D. McAllester, M. Collins, and F. Pereira. Case-factor diagrams for structured probabilistic modeling. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2004.
17. A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the International Conference on Machine Learning*, 2000.
18. R. McDonald and F. Pereira. Identifying gene and protein mentions in text using conditional random fields. In *Proceedings of the BioCreative Workshop*, 2004.
19. I. Muslea, C. Kloblock, and S. Minton. Active + semi-supervised learning = robust multi-view learning. In *Proc. of the International Conf. on Machine Learning*, 2002.
20. K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of Information and Knowledge Management*, 2000.
21. B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems*, 2003.
22. I. Tschantaris, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning*, 2004.
23. D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of the Annual Meeting of the Association for Comp. Ling.*, 1995.